

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A network management system, comprising:
 - a management system kernel that provides management systems with a run-time environment; and
 - a managed object generation environment that provides a development environment for managing applications, wherein the management system kernel can at least one of dynamically add and dynamically modify managed object (MO) information based upon an external meta file (EMM) from the managed object generation environment without interrupting an operation of the network management system.
2. (Original) The system of claim 1, wherein the management system kernel comprises:
 - a communication module that provides communication with a network manager;
 - a managed object framework that maintains information on MO classes;
 - a kernel that stores a dynamic class loading module and initializes the network management system, wherein said kernel establishes an association with other management systems through the communication module, performs management operations on MOs, adds the MO information in the managed object framework using the dynamic class loading module

and modifies the MO information in the managed object framework using the dynamic class loading module; and

a containment tree that organizes MO instances according to the information on MO classes and allows access to the MO instances when a management operation is performed in the network managed system.

3. (Original) The system of claim 2, wherein managed object framework maintains information on MO classes by registering MO class codes on a class information table.

4. (Original) The system of claim 2, wherein the kernel creates at least one dedicated agent to perform subsequent management operations from management systems with which an association has been established.

5. (Original) The system of claim 1, wherein the managed object generation environment comprises:

a MO compiler that compiles a MO script to generate the EMM file and MO class codes; and

a dynamic library storing the MO class codes.

6. (Original) The system of claim 5, wherein the EMM file includes MO class definition described in the MO script, and identifies a location and name in the dynamic library of a corresponding MO class.

7. (Original) The system of claim 5, wherein the MO class codes are compiled and stored in the dynamic library in a form of a dynamic link library.

8. (Previously Presented) A network management method comprising:

- (a) storing a dynamic class loading routine in a management system kernel;
- (b) initializing a managed system by constructing a managed object framework of the management system kernel that contains information of managed object (MO) classes;
- (c) creating MO instances and registering the MO instances in a containment tree of the management system kernel according to the information of MO classes;
- (d) checking whether a dynamic class loading flag is on when receiving a management operation request from a management system; and
- (e) updating MO information on the management system kernel without interrupting an operation of the management system by,
 - waiting for all threads to complete execution,
 - loading a dynamic library to the managed object framework utilizing the dynamic class loading routine when the dynamic class loading flag is on, and
 - resetting the dynamic class loading flag to off.

9. (Original) The method of claim 8, further comprising:

(f) performing the requested management operation and sending a management operation result to the management system requesting the management operation when the dynamic class loading flag is not on.

10. (Original) The method of claim 8, wherein dynamic class loading routine of (e) comprises:

opening a EMM file stored outside the management system kernel; and
loading the dynamic library indicated by the EMM file.

11. (Original) The method of claim 10, further comprising storing information about the management system requesting a management operation.

12. (Original) The method of claim 10, further comprising:

checking whether an additional thread can be created;

creating a dedicated agent to take charge of subsequent management operations from the management system requesting an association if an additional thread can be created; and

executing the dedicated agent thread and delivering association and management operation information to the dedicated agent to be utilized in interacting with the management system.

13. (Original) The method of claim 8, wherein the dynamic class loading flag is set on when the management system requesting a management operation invokes the dynamic class loading function to perform one of adding and modifying MO information in the management system kernel.

14. (Original) The method of claim 13, wherein the management system requesting the management operation invokes the dynamic class loading function by sending a control signal.

15. (Previously Presented) A network management method, comprising:
storing a dynamic class loading routine in a management system kernel of the managed system;

updating the management system kernel by modifying managed object (MO) information in the management system kernel while the managed system is operating by utilizing the dynamic class loading routine; and

generating the MO information to be modified and generating a external meta file (EMM) in a managed object generation environment of the managed system wherein the dynamic class loading routine opens the EMM file to modify the MO information in the management system kernel.

16. (Canceled).

17. (Currently Amended) The method of claim [[16]] 15, wherein the MO information to be modified is stored in the managed object generation environment in the form of a dynamic link library.

c'
18. (Original) A method of claim 17, wherein the EMM indicates an address of a dynamic link library corresponding to the MO information to be modified, and wherein the MO information is modified in the management system kernel according to said address.
